A Differentiated Message Delivery Architecture to Control Spam

Zhenhai Duan
Computer Science Department
Florida State University
Email: duan@cs.fsu.edu

Yingfei Dong
Dept. of Electrical Engineering
University of Hawaii
Email: yingfei@hawaii.edu

Kartik Gopalan
Computer Science Department
Florida State University
Email: kartik@cs.fsu.edu

Abstract

Unsolicited bulk electronic mail (spam) is increasingly plaguing the Internet Email system and deteriorating its value as a convenient communication tools. In this paper we argue that the difficulties in controlling spam can be attributed to the lack of receiver control on how different Email messages should be delievered on the Internet. In the current Email delivery architecture, a user can send messages to another at will, regardless of whether or not the latter is willing to accept the message. Based on this observation, we propose a differentiated message delivery architecture—DiffMail. In DiffMail, a user can classify Email senders into multiple classes and handle messages from each class differently. For example, although a receiver may directly accept messages from the regular correspondents, he may selectively ask other senders to store messages on the senders' *own mail servers, and pull the messages only if and when he wants to. In this paper we present the DiffMail architecture and illustrate some of the appealing advantages using real-world Email archives.*

1. Introduction

Electronic mail (Email) is one of the most popular applications on the Internet. However, the current Email delivery infrastructure provides an essentially effort-free platform for spammers to send a deluge of unsolicited commercial messages, commonly known by the term *spam*. Email spam is increasingly plaguing the Internet Email system and threatening to deteriorate its value as a convenient communication tools, as handling such messages consumes valuable user resources and time. Given the importance of controlling spam for preserving the value of the Email system, this issue has attracted a great amount of attention in both networking research and industrial communities. Many different spam control schemes have been proposed, and some of them have been deployed on the Internet, including various spam filters [8, 13, 15], sender authentication schemes [5, 12], and sender-discouragement mechanisms (to increase the cost of sending Email messages) [10, 14], such as paid mail. On the other hand, despite these anti-spam research and development efforts, the proportion of Email spam seen on the Internet has been continuously increasing in recent years. It is estimated that nowadays spam messages constitute 79% of all business Emails, up from 68% since the US federal Can-Spam Act of 2003 took effect in January 2004 [3].

In this paper we argue that the difficulties in restraining spam can be attributed to the lack of receiver[1] control on how Email messages are delivered. In the current Email delivery architecture, a user can send an Email to another at will, regardless of whether or not the receiver is willing to accept the message. In the early days of the Internet development, this was not a big problem as people on the network largely trusted each other. However, since the commercialization of the Internet in mid-1990, the nature of the Internet community has changed. It has become less trustworthy, and Email spam is possibly one of the most notable examples of the untrustworthy nature of the Internet [2]. To address this issue, we propose a novel differentiated message delivery architecture—DiffMail, which enables receivers to regulate message delivery. DiffMail relies on three important design notions: 1) **receiver-defined sender classification**. In DiffMail, a receiver can classify Email senders into different classes, based on, for example, how well he trusts

[1] Throughout the paper, a user (sender/receiver) can be either a Mail Transfer Agent [11] or a real Email user.

the senders. Given that receivers have more reliable knowledge about whom they want to communicate with, this provides us with a reliable, effective, and efficient way to classify messages; 2) **differentiated message deliveries**. Instead of allowing arbitrary senders to push messages to a receiver, the receiver may differentiate message deliveries based on the nature of corresponding senders, i.e., to which classes the senders belong. 3) **shift of storage and management responsibility from receivers to senders**. In DiffMail, non-regular correspondents need to store and manage their outgoing messages before these messages are retrieved by receivers. DiffMail has several salient advantages in restraining spam while preserving Email as an open and convenient communication tool. For example, in DiffMail, spammers have less flexibility to move around on the Internet (which will facilitate the design of spammer blacklists [13]), less incentive to recruit zombie machines to send spam, and greater responsibility for storing and managing the outgoing messages on their own mail servers.

The remainder of the paper is structured as follows. In Section 2 we present the basic DiffMail architecture, we illustrate some of the advantages using real-world Email archives in Section 3. We summarize the paper and discuss our ongoing work in Section 4.

2. DiffMail Architecture

In this section we present the essential components of the proposed DiffMail architecture and discuss their interactions (see Figure 1).

Receiver-defined sender classification: As discussed above, receivers should have more control over how they want to communicate with others in order to effectively fight Email spam. As a critical step towards achieving this goal, receivers must have a means to classify message senders. Based on this classification, different communication policies can be applied. To this end, DiffMail provides users with the capability of classifying senders into different classes. In DiffMail, senders are identified by a sender identifier **sid**. A **sid** can be defined at different granularities. In this paper we consider three granularities: Email accounts (in the form of *useraccount@domainname*), IP addresses, and network domain names. An Email account uniquely identifies a single sender; an IP address designates all senders whose Email messages are sent from this IP address; and a network domain name represents all senders within this domain.

We envision that the number of sender classes and the members of each class are a local matter, being managed by each individual network domain. To illustrate how senders may be classified, we present an example **sid** classification, which has three classes as follows:

- *well-known spammers*: This class contains all the **sid**s of the known spammers.
- *regular contacts*: The **sid**s of all the regular correspondents of a user are in this class.
- *unclassified sources*: This class contains all the senders that belong to neither the *regular contacts* nor the *well-known spammers* classes.

Receiver-defined sender classification lays the foundation for the DiffMail architecture. First, this sender classification is defined by users instead of automated spam filters, and users certainly have more reliable knowledge about whom they want to communicate with than spam filters. Therefore, the receiver-defined sender classification provides us with a more reliable, effective, and efficient way to classify messages. Second, based on this classification, different communication policies can be applied to different classes in order to differentiate message delivery to restrain spam. For such a sender classification to work properly, we must prevent spammers from impersonating other users. As a starting point, existing sender authentication schemes such as Sender-ID and DomainKeys can be used [12, 5] for this purpose.

Differentiated message deliveries–sender push vs. receiver pull: The value of the Email system largely relies on its *convenient, open, and cost effective* communication model. In the same time, it also presents great challenges for us to fight spam. For example, one proposed method to restrict spam is to increase the effort that takes to send a message. However, it makes Email less convenient, as it not only affects spammers but also regular correspondents. On the other hand, a closed communication model such as the one supported in Instant Messengers (IM) provides possibly the best means to fight spam. However, this is not suitable for Email as its value depends on its openness.

To effectively address these challenging issues, we propose and advocate a differentiated message delivery model, based on the nature of message senders. In the following, we outline one of the possibly realizations of such a model, DiffMail. In essence, DiffMail allows users to differentiate message delivery for distinct sender classes, as defined above. DiffMail delivers messages from the *regular contact* class in the same way as in the current Email architecture, in particular, complete messages (including both headers and bodies) are delivered (*pushed*) directly from a sender to a receiver. On the other hand, DiffMail will not accept any messages from the *well-known spammers* class, i.e., messages from the well-know spammers class are never delivered to receivers.

Unlike messages from regular contacts or well-known spammers, the ones from *unclassified sources* can be either spam or regular messages. Hence this represents the most critical category to manage in effectively controlling

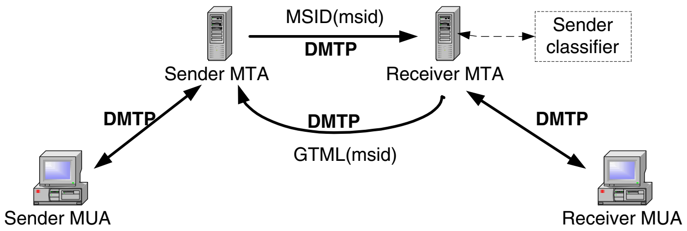

Figure 1. Illustration of the DiffMail architecture.

spam. To discourage spammers we should prevent their messages from being delivered to receivers. Furthermore, we should also provide legitimate users (who are not in the regular contact class yet) with a way to express the intention to communicate. To balance these two considerations, DiffMail only accepts the envelopes (or headers) of messages from this class [11]. The complete messages need to be stored at the *sender Mail Transfer Agents (MTAs)*. If a receiver wants to read such messages, he can retrieve (or *pull*) the messages from the sender MTAs at a later convenient time, using an extended version of the Simple Mail Transfer Protocol (SMTP) [11]. We refer to this extended version of SMTP as Differentiated Mail Transfer Protocol (DMTP) [6]. DMTP extends SMTP in two aspects: it allows senders to manage their outgoing message folders (see below), and it supports message retrieval by receivers. As a summary, we see that in DiffMail, an arbitrary sender cannot, at will, *push* a message to a receiver. Instead, the receiver has the control over whether or not he will *pull* the message from the sender. Given the rapid bandwidth and server capacity expansion on the Internet, retrieving messages from sender mail servers should not affect the users' perceived message reading experience.

We note that there is a fundamental difference between message pull in DiffMail and URL embedded in many current spam messages. The address in the URL is normally not related to the sending machine of the message, which makes it hard to identify the one who is responsible for the spam. On the other hand, messages in DiffMail have to be stored on the sender mail servers instead of third-party machines before they are retrieved. In this way, we obtain several advantages in restricting spam, as will be shown in the next section.

Responsibilities of senders and receivers: A sender uses some Mail User Agent (MUA) to compose the messages that he wants to send [11]. After a message is composed by the sender, the sender delivers the message to the sender Mail Transfer Agent (MTA) using DMTP. For simplicity, we refer to a sender MTA server as an SMTA, and a receiver MTA server as an RMTA.

All the outgoing messages are stored at the SMTA. For this purpose, the SMTA maintains an outgoing message folder for each sender. This folder contains all user's messages that have not been delivered and have not been deleted. A user can explicitly delete his outgoing messages from the SMTA folder by means of DMTP. An SMTA can also delete a message on behalf of users, after the message has been delivered to *all* the intended receivers or after a certain user-configurable expiry time. Therefore senders in DiffMail have more responsibility to store and manage their outgoing Email messages compared with that in the current Email architecture. This is especially true for spammers, who are most likely in the well-known spammer classes or the unclassified classes of others. For both these cases, the senders need to store and manage their undelivered messages on their own MTA servers.

An SMTA communicates with an RMTA using DMTP, trying to deliver messages from the sender to the receiver. A receiver differentiates the treatment of messages from different senders by defining a number of sender classes or groups. We refer to this functionality module as *sender classifier* (See Figure 1). We will discuss the overhead of managing the sender classifier in the next section. Messages from regular contacts and unclassified sources are stored in different folders so that receivers may not spend time on messages from unclassified sources (note that such messages only contain the header information). If a receiver indeed wants to read a complete message from an unclassified source, he will inform his own RMTA, and the RMTA will retrieve the message from the SMTA on behalf of the receiver. After the message has been pulled to the RMTA, conventional virus/worm scanning tools and content-based spam filter can be applied to further alert the receiver on potential virus or spam. Therefore, *DiffMail does not exclude the usages of existing Email protection schemes.*

3. Advantages of the DiffMail Architecture

Less flexibility for spammers to move around: By asking senders (non-regular contacts) to maintain messages on their own mail servers, spammers are forced to keep their servers up. They cannot simply send a large number of spam messages, shut down their servers, and switch to another domain (and/or change IP addresses). They need to wait

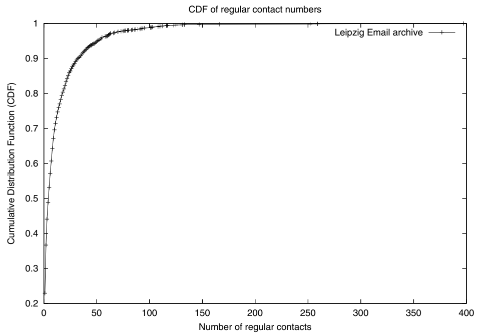

Figure 2. CDF of regular contact numbers.

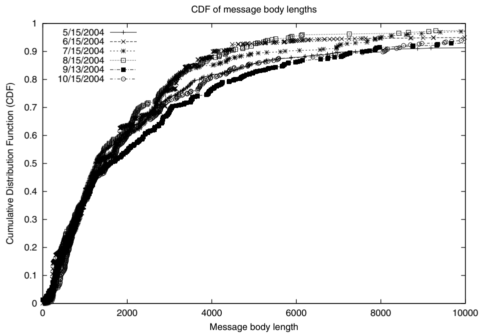

Figure 3. CDF of message body lengths.

for receivers to retrieve the messages. This helps to identify spammers and improve the effectiveness of IP-address based filtering schemes such as RBL [13]. If spammers can frequently change their IP addresses or domain names, the effectiveness of such schemes will be limited.

Less incentive for spammers to recruit zombie machines to send spam: As the networking community is fighting back on spam, spammers are more and more relying on hacked zombie machines to send spam. A recent study reported that about 80% of spam were sent from such zombie machines [9]. Although there is a theoretically possibility for hackers to crack major mail servers, the vast majority of zombie machines are home user machines, as major mail servers are better maintained and protected. After breaking into home user machine, a spammer can use the hacked system to send a great amount of spam to arbitrary receivers. If DiffMail is deployed, spammer can only successively send spam to the receivers who have included the owner of the hacked system in the regular contact class. Therefore, DiffMail may reduce the incentive for spammers to use zombie machines to send spam, as there is a cost to recruit any zombie machine for hackers.

To have a more concrete idea on this, we conduct a simple empirical study on the number of regular contacts that a user may have. For this study, we use the Email log from the University of Leipzig, Germany [7], which contains Emails in and out of the university from 9/2/2001 to 11/28/2001. The log records the following information of a message: date, time, anonymized sender and receiver addresses. Addresses are also distinguished as being "internal" or "external" depending on whether it is within the university domain. This data set contains a total of $447,543$ messages. We refer to this date set as "Leipzig Email Archive."

In the Leipzig Email Archive data set, we are not able to distinguish messages from spammers and those from regular contacts. Because of this, we only consider the addresses to which a user sent messages and regard this set of addresses as regular contacts of the user. Moreover, we only compute regular contacts for users who belong to the university (i.e., internal users). We observe $1,753$ internal users who sent messages in the data set. Figure 2 presents the cumulative distribution function (CDF) of the number of regular contacts for such users. From the figure we can see that more than 90% of users have less than 35 regular contacts[2]. Given the limited data we have, this may only represent a lower bound on the number of regular contacts for the users. However, we conjecture that a common user on the Internet may not have much more regular contacts than what we obtained from the data. Consequently, a spammer in general can only successively deliver spam to several tens/hundreds of receivers, after a hacked home user machine is recruited.

Resource consumption paradigm shift and incentive for networks to deploy: In DiffMail, senders have greater responsibility in sending a message than in the current Email architecture. They need to store and manage messages on their own mail servers before the messages are retrieved by the receivers. A message that has not been delivered over a certain period of time needs to be deleted by the SMTA or by the sender himself. On the other hand, the implication for receivers is two-fold. First, by only delivering the envelope of a message from sender to receiver, less bandwidth, storage, and time will be occupied at the receiver side, which may be especially important for dial-up users. On the other hand, if the user indeed wants to read the message from an unclassified source, extra bandwidth and time will be used. However, users will most unlikely be interested in messages from unclassified sources, and therefore, the majority of such messages will not be retrieved. More importantly, if the majority of messages from unclassified sources are not delivered, much less bandwidth on the Internet as a whole will be consumed by spam. To have a better understanding on this, we conduct another simple empirical study on the lengths of spam messages. We use the data from the Spam Archive site [1]. This site maintains archives of Email spam contributed by Internet Email users. We (randomly) select spam archived by the site on

[2] Note that the data was collected only over 90 days in 2001.

Messages	5/15	6/15	7/15	8/15	9/13	10/15
Complete	478	2742	415	346	462	393
Damaged	4	2	0	2	1	5
Total	482	2744	415	348	463	398

Table 1. Number of messages in each Spam Archives

5/15/2004, 6/15/2004, 7/15/2004, 8/15/2004, 9/13/2004 (no archives on 15th and 14th of the month), and 10/15/2004. Due to forwarding problems, some messages in the archives are damaged or incomplete, we remove such messages from the data sets before we analyze the data. Table 1 shows the total number of messages and the number of complete messages. We refer to the data set *after excluding the incomplete messages* as a "Spam Archive."

Figure 3 depicts the CDFs of spam message body lengths. (Note that we exclude message headers when we compute this length.) We can see from the figure that more than half of spam have a body longer than 1.5 KB, and 60% of spam longer than 2 KB. Although the bandwidth and storage saving from not retrieving a single message may not be significant, the overall potential savings will be promising considering the massive volume of spam on the Internet.

DiffMail is not free. Receivers or RMTAs need to maintain sender classifiers. However, as we showed above, a normal user only has several tens/hundreds of regular contacts, therefore the cost to maintain this classifier will be trivial. Moreover, this classifier is not changed frequently. Lastly, as we will show in the next section when we present the design of DiffMail, it is not necessary for end users to maintain sender classifiers for DiffMail to work effectively. In a nutshell, networks will only need trivial efforts to support DiffMail to protect themselves from spam. Moreover, DiffMail is *incrementally deployable* (due to the page limit, we will not discuss this in detail. Interested readers are referred to [6]). Since it does not require all mail servers to support DMTP to work effectively, DiffMail provides adequate incentives for networks to deploy it.

Preserving Email as an open and convenient communication tool: In DiffMail, messages from regular contacts are handled in the same way as in the current Email architecture. Hence no extra efforts are needed for sending such messages. This will preserve Email as a convenient communication tool. On the other hand, although messages from unclassified sources cannot be directly delivered, DiffMail provides a means for such senders to express the intention to communicate. In this way, we can retain Email as an open and generic communication tool.

4. Conclusion and Ongoing Work

In this paper we proposed and studied a differentiated messsage delivery architecture, DiffMail, to control Email spam on the Internet. In DiffMail, a user can classify Email senders into multiple classes and handle messages from each class differently. For example, although a receiver may directly accept messages from the regular correspondents, he may selectively ask other senders to store messages on the *senders'* own mail servers, and pull the messages only if and when he wants to. In this paper, we also used real-world Email archieves to illustrate the advantages of the proposed architecture. Currently we are developing a prototype of DiffMail based on Sendmail [4].

References

[1] S. Archive. Donate your spam to science. http://www.spamarchive.org/.

[2] M. Blumenthal and D. Clark. Rethinking the design of the internet: the end-to-end arguments vs. the brave new world. *ACM Transactions on Internet Technology*, 1(1), Aug. 2001.

[3] T. Claburn. Big guns aim at spam. *Information Week*, Mar. 2004.

[4] S. Consortium. Welcome to sendmail.org. http://www.sendmail.org/.

[5] M. Delany. Domain-based email authentication using public-keys avertised in the DNS (domainkeys). Internet Draft, Aug. 2004. Work in Progress.

[6] Z. Duan, Y. Dong, and K. Gopalan. Diffmail: A differentiated message delivery architecture to control spam. Technical Report TR-041025, Department of Computer Science, Florida State University, Oct. 2004.

[7] H. Ebel. The data of the e-mail network. http://www.theo-physik.uni-kiel.de/ ebel/email-net/email%5Fnet.html.

[8] P. Graham. A plan for spam. *http://www.paulgraham.com/spam.html*, January 2003.

[9] S. Incorporated. Trend analysis: Spam trojans and their impact on broadband service providers, June 2004.

[10] A. Juels and J. Brainard. Client puzzles: A cryptographic defense against connection depletion attacks. In *Proceedings of NDSS-1999 (Networks and Distributed Security Systems)*, Feb. 1999.

[11] J. Klensin. Simple mail transfer protocol. RFC 2821, Apr. 2001.

[12] J. Lyon and M. Wong. Sender ID: Authenticating e-mail. Internet Draft, Aug. 2004. Work in Progress.

[13] RBL. Real-time spam black lists (rbl). http://www.email-policy.com/Spam-black-lists.htm.

[14] V. Rishi. Free lunch ends: e-mail to go paid. *The Economic Times*, Feb. 2004.

[15] SpamAssassin. The apache spamassassin project. http://spamassassin.apache.org/.